

User Manual
for
CG300-SW-104P

300 MHz PC/104-Plus
Frequency Synthesizer
(with Sweep Capability)

CHASE SCIENTIFIC COMPANY
P.O. Box 1487
Langley, WA 98260

Email: techsupport@chase2000.com
Web: <http://www.chase2000.com>

Original Document: cg300-SW-104p_manual.odt
(created 12-09-2008, updated 03/25/12)

© Copyright 2008-2012 by
Chase Scientific Company

This manual, the CG300 module, and the software drivers outlined in this document are copyrighted with all rights reserved. Under the copyright laws, the above mentioned may not be copied, in whole or in part, without the express written consent of Chase Scientific Company.

TABLE OF CONTENTS

1 GENERAL INFORMATION.....	4
1.1 INTRODUCTION.....	4
1.2 REFERENCES.....	4
1.3 DELIVERABLES.....	4
1.3.1 Software.....	4
1.3.2 Hardware.....	5
1.3.3 Checklist.....	5
1.4 PRODUCT SPECIFICATION.....	6
1.5 TECHNICAL SUPPORT / SOFTWARE UPDATES.....	6
1.6 WARRANTY.....	7
2 HARDWARE DESCRIPTION.....	8
2.1 INTRODUCTION.....	8
2.2 BLOCK DIAGRAM.....	8
2.3 BOARD DRAWING.....	9
2.4 BOARD I/O.....	10
2.4.1 Header Pinouts and Pin Descriptions.....	10
2.5 HARDWARE REGISTER DEFINITIONS.....	10
3 THEORY OF OPERATION.....	11
3.1 INTRODUCTION.....	11
4 SOFTWARE DRIVERS.....	11
4.1 INTRODUCTION.....	11
4.2 DRIVER INSTALLATION.....	11
4.2.1 Windows 98/ME/2000/XP.....	11
4.2.2 Windows 7 (TBD).....	12
4.3 FUNCTION CALLS.....	12
4.3.1 Function Declarations in C.....	12
4.3.2 Function Call Descriptions / Usage.....	12
4.3.2.1 cg300_CountCards().....	12
4.3.2.2 cg300_Open().....	13
4.3.2.3 cg300_Close().....	13
4.3.2.4 cg300_initialize();.....	14
4.3.2.5 cg300_SetFrequency().....	14
4.3.2.6 cg300_SetSweep().....	15
4.4 PROGRAMMING EXAMPLES.....	15
4.4.1 Generic C Snippets.....	15
5 MISCELLANEOUS.....	17
5.1 CALIBRATION.....	17
5.2 MAINTENANCE.....	17

ILLUSTRATIONS / TABLES

FIGURE 1 – BLOCK DIAGRAM.....8

FIGURE 2 – BOARD LAYOUT.....9

FIGURE 3 - 4-PIN DIGITAL I/O CONNECTOR.....10

1 GENERAL INFORMATION

1.1 Introduction

The CG300 is a PC/104-Plus based Clock Synthesizer with maximum output frequencies of 300 MHz. Each board has as standard the following features:

- (1) 1 Vpp Sinewave output, AC coupled through a transformer.
- (2) 3.3V TTL outputs, one normal / one inverted.
- (1) 3.3V TTL Marker output used to indicate beginning/ending of Sweep Mode Only.

The CG300 have a crystal based on-board clock sources which provides several orders of magnitude better phase noise than a PLL based design.

NOTE: The model number CG300 is used in the following paragraphs represent CG300-SW-104P. This was done to improve readability of the manual.

1.2 References

PC/104-Plus Specification, Version 2.0, November 2003 by PC/104 Embedded Consortium.

“P996.1 Standard for Compact Embedded-PC Modules”; PC/104 Specification, Version 2.3, June 1996 by PC/104 Consortium.

“IEEE P996 draft standard”, 1987 by IEEE Standards Committee. This bus standard was the basis for the first PC/104 standard released in 1992.

1.3 Deliverables

1.3.1 Software

The CG300 comes with drivers for **Windows 98/NT/2000/XP**. Software can be downloaded from “www.chase2000.com”. Call Chase Scientific for the latest information on drivers for other operating system platforms or check the web site for the latest updates.

Software drivers are provided as a Dynamic Link Library (*.DLL) which is compatible with most 32-bit windows based development software including Microsoft C/C++, Borland C/C++, and Borland Delphi. This DLL uses the “**cdecl**” calling convention which is default compatible with the compilers above and provides easy to use function calls to the system drivers “windrvr6.vxd” for Windows NT and “windrvr6.sys” for Windows 98/2000/XP.

Actual Listing of files in Zip File for Windows:

```

----- D I R E C T O R I E S / F I L E S -----
BASE_DIR
|
|  readme.txt                // This file.
|  cg300-sw-104p_drawing.pdf // Reference drawing for CG300 (connector descriptions)
|
|  Register_cg300_Win98_2000_XP.bat // Installs Kernel driver for Win2000/XP
|  UnRegister_cg300_Win98_2000_XP.bat // Uninstalls Kernel driver for Win2000/XP
|
|  Register_cg300_NT4.bat // Installs Kernel driver for Windows NT4.0
|  UnRegister_cg300_NT4.bat // Uninstalls Kernel driver for Windows NT4.0
|
|  wdregl16.exe // Called by Register_cg300_Win98_2000_XP.bat
|  wdreg.exe // Called by Register_cg300_NT4.bat
|  windrvr6.inf // Setup information file automatically called by above exe(s).
|
|  cg300_dll.dll // DLL for 98/ME/NT4/2000/XP (extern "C" _declspec(dllexport)
|  )
|  cg300_dll_import.h // Header file for DLL
|  cg300_dll.lib // Library file for DLL in Borland C++
|
|  cg300_dll_msvc.lib // LIB file for MSVC Project to compile DLL above
|  example_snippet.txt // Example of function calls
|
|  cg300_diag.exe // Simple GUI to test DLL and Kernel drivers
|
|  CG300_PC104P.inf // Plug-And-Play file needed by 98/ME/NT4/2000/XP for automatic
|  // hardware configuration.
|
|  windrvr6.sys // Windows 98/ME/2000/XP Driver - copy this virtual driver
|  // to "c:\<windir>\system32\drivers\" if not automatically done
|  // so after running batch file.
|
|  windrvr6.vxd // Windows NT4.0 Driver
|
----- E N D -----

```

1.3.2 Hardware

The CG300 hardware consists of a single PC/104-Plus module. The module is shipped with a manual with complete hardware and software descriptions. This card can only be accessed as a PCI device according to PCI Local Bus Specification.

1.3.3 Checklist

Item #	Qty	Part Number	Description
1	1	CG300-PCI	100 KHz - 300 MHz PCI Clock Synthesizer Card
2	1	CG300-Drivers	Downloadable from "www.chase2000.com". Includes Dynamic Link Libraries for Windows 98/NT/2000/XP, manual, debug GUI, and examples.
3	1	CG300-Manual	User manual for CG300.

1.4 Product Specification

(all specifications are at 25C unless otherwise specified)

I/O SPECIFICATIONS

Sine Output	
Output Voltage	1.0 Vpp Sinewave (typical)
Frequency Range	100KHz to 300 MHz
Frequency Step Size	0.23 Hz
Sweep Rate (time/step)	16 ns < SwTime < TBD
TTL	
	(2) 3.3V TTL Clock output. Frequency ranges same as Sine Out. One Normal, one Inverted. Tr/Tf = 1ns Typical
	(1) Marker Output
Clock Characteristics	
Phase Noise	-100dBc/Hz @ 1KHz Offset Typical @ Fc= 100 MHz [CG300]
Jitter	< 25 picoseconds [bandwidth from 1KHz to Fmax]

GENERAL

Power Supply (Vcc)	+ 5.0V +/- 10%
Operating Temperature	0 to 70 degrees C standard -40 to +85 degrees C extended
Operating Humidity	5 to 95% non-condensing
Size	PC104-Plus
Data Bus	32-bits PC/104-Plus (ISA Bus is pass-thru only, open circuit)

1.5 Technical Support / Software Updates

For technical support:

Email	techsupport@chase2000.com
Mail	Chase Scientific Company P.O. Box 1487 Langley, WA 98260

For software updates:

Email	techsupport@chase2000.com
Web	http://www.chase2000.com

1.6 Warranty

Chase Scientific Company (hereafter called Chase Scientific) warrants to the original purchaser that its CG300-SW-104P, and the component parts thereof, will be free from defects in workmanship and materials for a period of ONE YEAR from the date of purchase.

Chase Scientific will, without charge, repair or replace at its option, defective or component parts upon delivery to Chase Scientific's service department within the warranty period accompanied by proof of purchase date in the form of a sales receipt.

EXCLUSIONS: This warranty does not apply in the event of misuse or abuse of the product or as a result of unauthorized alterations or repairs. It is void if the serial number is altered, defaced or removed.

Chase Scientific shall not be liable for any consequential damages, including without limitation damages resulting from loss of use. Some states do not allow limitation or incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This warranty gives you specific rights. You may also have other rights that vary from state to state.

Chase Scientific warrants products sold only in the USA and Canada. In countries other than the USA, each distributor warrants the Chase Scientific products that it sells.

NOTICE: Chase Scientific reserves the right to make changes and/or improvements in the product(s) described in this manual at any time without notice.

2 HARDWARE DESCRIPTION

2.1 Introduction

The CG300 hardware consists of the following major functions:

- (1) Sinewave Output [SMA]
- (2) 3.3V TTL Outputs (+/- Comparator)
- (1) 3.3V Marker Output (for Sweep Mode Only)

2.2 Block Diagram

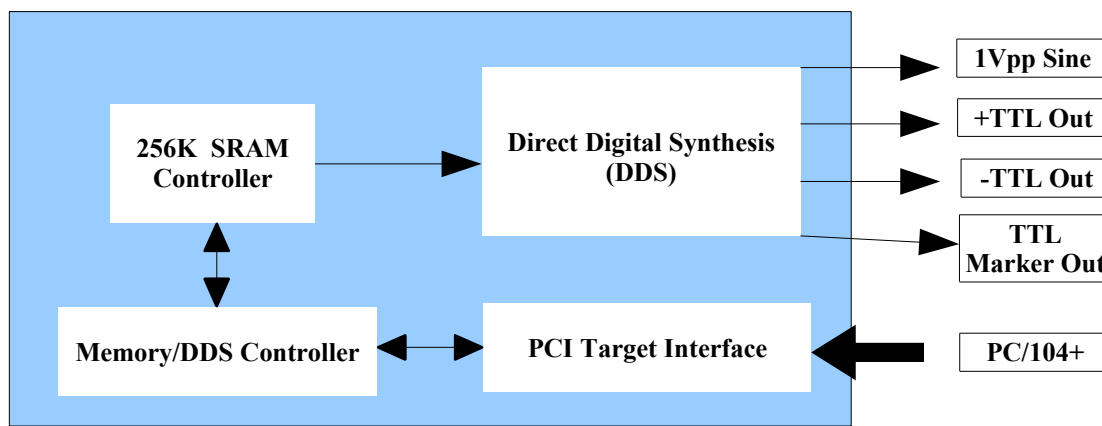
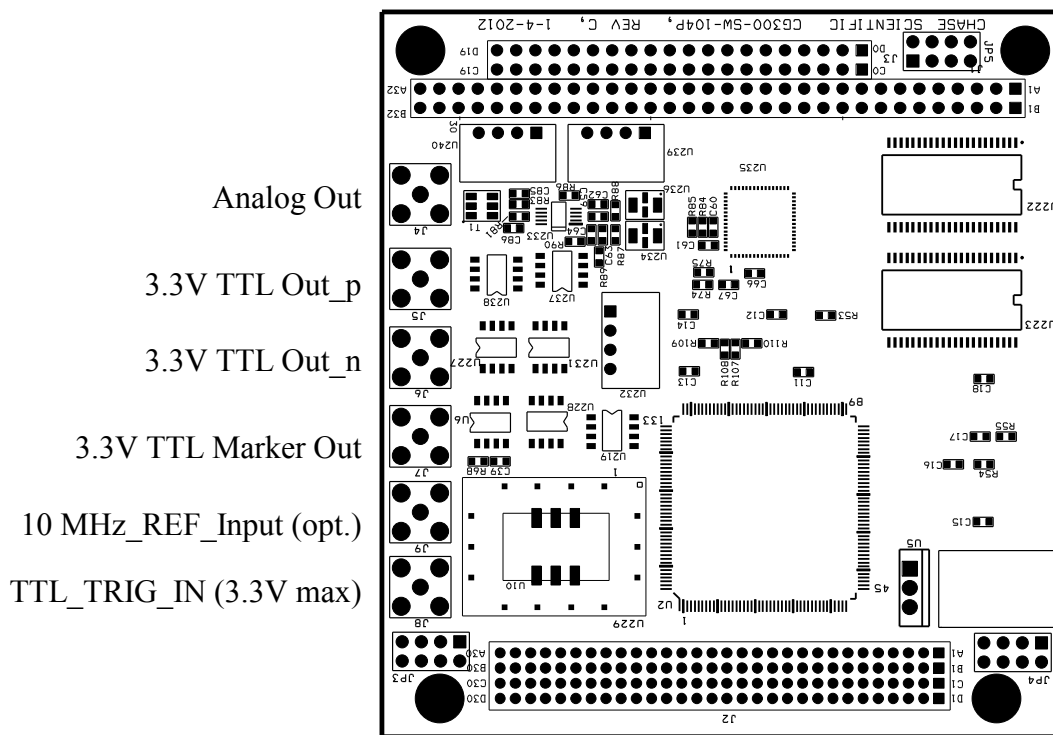


Figure 1 – Block Diagram

2.3 Board Drawing



2.4 Board I/O

2.4.1 Header Pinouts and Pin Descriptions

N/A

2.5 Hardware Register Definitions

Due to hardware protocol complexity, the details on how to use these ports are beyond the scope of this manual. The software drivers should provide an effortless integration path for the user. However, if the need arises that is beyond what the software drivers can provide, please call the factory for technical support.

3 THEORY OF OPERATION

3.1 Introduction

The CG300 is primarily comprised of a PCI target controller, a Direct Digital Synthesis (DDS) IC, SRAM for sweep data, filtering, and output buffers. Outputs include a 1Vpp sinewave output (transformer coupled) and (2) 3.3V TTL outputs (+/-). All outputs have SMA connectors.

Software controls all static and sweeping frequency values as well as sweep durations. Although the drivers are primarily designed for software developers, we have included a GUI to exercise the card and verify that the low level drivers are installed properly.

4 SOFTWARE DRIVERS

4.1 Introduction

The primary objective in designing software drivers is to get the user up and running as quickly as possible. The details on installing software drivers are listed in 4.2.xx. While the listing of function calls and their parameter definitions are listed in section 4.3.xx, the programming examples in section 4.4.x will show you how to include them into your programs.

The drivers are designed to work under Windows 98/ME/NT4/2000/XP.

4.2 Driver Installation

4.2.1 Windows 98/ME/2000/XP

- 1) Do not install CG300 card at this time.
- 2) UnZip all files into directory "C:\temp\CG300\" (create directories if needed) You can move and/or copy the files later to a directory of your choice.
- 3) Run Register_CG300_Win98_2000_XP.bat. This will copy the Kernel driver windrvr6.sys to "c:\<windir>\system32\drivers\" directory and will register the Kernel driver in the Windows Registry so that it starts up each time the computer is rebooted.
- 4) Power off computer. Insert CG300 card. Power up computer.
- 5) When OS asks for Driver File point to "CG300_PCI.inf". If OS does not ask for file, then check hardware configuration and update if not listed properly under "Jungo" in Device Manager (see below).

To check to see which driver is installed, do the following:

=> Control Panel

```

=> System
=> Hardware
=> Device Manager
=> Jungo
    CG300_PC104P
    WinDriver

```

If you see another driver in place of "CG300_PC104P", then right click the first device under Jungo and click properties. Update the driver by pointing to "CG300_PC104P.inf". You may have to go through a series of menus.

4.2.2 Windows 7 (TBD)

4.3 Function Calls

4.3.1 Function Declarations in C

```

//-----
#ifndef cg300_dllH
#define cg300_dllH
//-----

#define IMPORT extern "C" __declspec(dllimport)

// USER FUNCTION CALLS ==>
IMPORT DWORD cg300_CountCards(void);
IMPORT DWORD cg300_Open(DWORD CardNum);
IMPORT DWORD cg300_Close(DWORD CardNum);

IMPORT void cg300_initialize(DWORD BrdNum);

IMPORT void cg300_SetFrequency(DWORD BrdNum, float DDS_Frequency);

IMPORT void cg300_Sweep(
    DWORD BrdNum,
    DWORD FREQ_START,
    DWORD FREQ_STOP,
    double SWEEP_TIME,
    double TOTAL_TIME,
    DWORD BLANK_FREQ
);

#endif

```

4.3.2 Function Call Descriptions / Usage

4.3.2.1 cg300_CountCards()

Description

Returns number of CG300 cards present on computer.

Declaration

```
DWORD cg300_CountCards(void);
```

Parameters

none

Return Value

Returns with an encoded value which represents the number of CG300.

Return Values:

- 0: Kernel Driver ok, but no card found
- 1-4: Normal range of possible number of cards detected
- 13: Kernel Driver not working

Example

```
DWORD Num_cg300_Boards = cg300_CountCards();
```

4.3.2.2 cg300_Open()**Description**

Loads the CG300 software drivers and sets the CG300 board to its default state.

Declaration

```
DWORD cg300_Open(DWORD CardNum);
```

Parameters

CardNum: 1 <= CardNum <= 4

Return Value

Returns with error code. A "0" means everything is fine. See below for details for other values.

Return Values:

- 0: Opened Windriver Successfully and CG300 Card Found Successfully
- 1: Opened Windriver Successfully, but NO CG300 CARDS FOUND
- 2: Opened Windriver Successfully, CG300 card found, but Cannot Open Driver to other Board Functions.
- 3: Opened Windriver Successfully, Board already open.
- 6: Card number exceeds number of cards.
- 13: FAILED TO OPEN Windriver Kernel Driver

Example

```
DWORD OpenErrorCode = cg300_Open(1); // Opens Board Number 1 and stores value.
```

4.3.2.3 cg300_Close()**Description**

Closes cg300 drivers. Should be called after finishing using the driver. However, if no other software uses the "windrv.xxx" (usual situation), then there is no need to close it until user is ready to completely exit from using their main software program.

Declaration

```
DWORD cg300_Close(DWORD CardNum);
```

Parameters

CardNum: 1 <= CardNum <= 4

Return Value

Returns with error code. A "0" means everything is fine. See below for details for other values.

Return Values:

- 0: Closed Windriver Successfully for CG300 card requested.
- 5: CG300 Card Already Closed for card requested.
- 13: FAILED TO ACCESS Windriver Kernel Driver

Example

```
DWORD CloseErrorCode = cg300_Close(1);
```

4.3.2.4 cg300_initialize();**Description**

Sets up DDS chip so that programming can be performed.

Declaration

```
void cg300_Reset (DWORD BrdNum);
```

Parameters

CardNum: 1 <= CardNum <= 4

Return Value

None.

Example

```
cg300_initialize(1);
```

4.3.2.5 cg300_SetFrequency()**Description**

Sets internal clock rate based on on-board clock or external clock (DivideBy=8).

Declaration

```
void cg300_SetFrequency (DWORD BrdNum, float DDS_Frequency);
```

Parameters

BrdNum: 1 <= CardNum <= 4

DDS_Frequency: DDS_Frequency <= (0.4)(Master Clock) [Master Clock = 1000000000]

Return Value

None.

Example

```
cg300_SetFrequency(1,100); // Set board #1 to 100 MHz (CG300)
```

4.3.2.6 cg300_SetSweep()

Description

Automatically creates a looping frequency sweep waveform based on start frequency, stop frequency, sweep time, total time, and frequency of blank time.

Declaration

```
void cg300_Sweep(
    DWORD BrdNum,
    DWORD FREQ_START,
    DWORD FREQ_STOP,
    double SWEEP_TIME,
    double TOTAL_TIME,
    DWORD BLANK_FREQ
);
```

Parameters

BrdNum:	1 <= CardNum <= 4	
FREQ_START	1 <= FREQ_START <= 400000000	
FREQ_STOP	1 <= FREQ_STOP <= 400000000	[must be greater than FREQ_START]
SWEEP_TIME	TBD <= SWEEP_TIME <= TBD	
TOTAL_TIME	TBD <= TOTAL_TIME <= TBD	
BLANK_FREQ	1 <= BLANK_FREQ <= 400000000	

Return Value

None.

Example

```
cg400_Sweep(1,          // BrdNum (DWORD)
    175000000,          // FREQ_START (DWORD)
    245000000,          // FREQ_STOP (DWORD)
    655e-6,             // SWEEP_TIME (double)
    5000e-6,            // TOTAL_TIME (double)
    10000000            // BLANK_FREQ (DWORD)
);
```

4.4 Programming Examples

4.4.1 Generic C Snippets

Standard Sequence of Function Calls ==>

To Set Fixed Frequency:

```
cg300_CountCards(); // Check to see if card is there
cg300_Open(1);      // Opens cg300 Kernel Driver Access

cg300_initialize(1); // Initializes card # 1
cg300_SetFrequency(1, 155520000.00); // Sets Frequency to 155.52 MHz

cg300_Close(1);     // Run this before quitting program
```

To Set Sweep Mode:

```
cg400_CountCards();    // Check to see if card is there
cg400_Open(1);         // Opens cg400 Kernel Driver Access

cg400_initialize(1);    // Initializes card # 1
cg400_Sweep(1,          // BrdNum  (DWORD)
            175000000,   // FREQ_START (DWORD)
            245000000,   // FREQ_STOP  (DWORD)
            655e-6,      // SWEEP_TIME  (double)
            5000e-6,      // TOTAL_TIME  (double)
            10000000     // BLANK_FREQ  (DWORD)
            );

cg400_Close(1);        // Run this before quitting program
```


5 MISCELLANEOUS

5.1 Calibration

The CG300 has no user feature to calibrate for input and output gain accuracy or offsets.

5.2 Maintenance

No maintenance is required. However, a yearly calibration is recommended if the user desires to maintain the CG300 modules' specified accuracy. Call factory for maintenance and/or extended warranty information.

Trademarks:

MS-DOS, Windows 3.1, Windows 95, Windows NT, Windows 2000, and Windows XP are registered trademarks of Microsoft Corporation.